Stateful Intrusion Detection for IEC 60870-5-104 SCADA Security

Yang, Y., McLaughlin, K., Sezer, S., Yuan, Y. B., & Huang, W. (2014). Stateful Intrusion Detection for IEC 60870-5-104 SCADA Security. In *2014 IEEE PES General Meeting Conference & Exposition* (pp. 1-5) https://doi.org/10.1109/PESGM.2014.6939218

Published in:
2014 IEEE PES General Meeting Conference & Exposition

Document Version:
Peer reviewed version

Stateful Intrusion Detection for IEC 60870-5-104 SCADA Security

Y. Yang[1,2], K. McLaughlin[2], S. Sezer[2], Y.B. Yuan[1], W. Huang[1]
(1. Jiangsu Electric Power Company Research Institute, Nanjing, China, yyang09@qub.ac.uk;
2. Centre for Secure Information Technologies (CSIT), Queen's University Belfast, Belfast, UK)

***Abstract*—Cyber threats in Supervisory Control and Data Acquisition (SCADA) systems have the potential to render physical damage and jeopardize power system operation, safety and stability. SCADA systems were originally designed with little consideration of escalating cyber threats and hence the problem of how to develop robust intrusion detection technologies to tailor the requirements of SCADA is an emerging topic and a big challenge. This paper proposes a stateful Intrusion Detection System (IDS) using a Deep Packet Inspection (DPI) method to improve the cyber-security of SCADA systems using the IEC 60870-5-104 protocol which is tailored for basic telecontrol communications. The proposed stateful protocol analysis approach is presented that is designed specifically for the IEC 60870-5-104 protocol. Finally, the novel intrusion detection approach are implemented and validated.**

***Index Terms*--SCADA, Cyber-security, Intrusion detection, IEC 60870-5-104.**

I. INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) systems have long played a critical role in power system operation and communications. The increasing standardization and interconnection of SCADA systems in Smart Grids potentially widens the prospect of cyber attacks from malicious sources. Furthermore, cyber-security aspects have not generally been considered during the design phase of most SCADA networks. Therefore, SCADA systems could be compromised by malicious attackers or disgruntled employees via unauthorized access at vulnerable points. Such intrusion has the potential to render simple or elaborate attacks which may jeopardize the system operation and which may ultimately lead to severe physical damage (e.g., the first-ever sophisticated control system malware, *Stuxnet* [1]). The marriage of new cyber technology to traditional control systems creates an environment where cyber and physical assets interact in ways never envisaged or planned for in most power control systems. Protecting SCADA systems from cyber threats is therefore a pertinent topic and of immediate relevance to modern power systems and smarter grids.

Currently, a number of open international standards exist in SCADA systems of the electrical power, such as Distributed Network Protocol Version 3 (DNP3), IEC 60870-5 series, and IEC 61850. For example, the IEC 60870-5-104 transmission protocol [2] presents network access for IEC 60870-5-101 [3] based on Transmission Control Protocol/Internet Protocol (TCP/IP), which can be utilized for basic telecontrol tasks in SCADA systems. However, the IEC 60870-5-104 protocol transmits messages in clear text without any authentication mechanism [4]. Furthermore, the IEC 60870-5-104 protocol is based on TCP/IP which also has cyber-security issues itself. (IEC/104 is used as the notation, instead of IEC 60870-5-104 in the remainder of the paper.)

In related research, the authors [4] have previously implemented signature-based and model-based detection approaches using *Snort* to improve the cyber-security of SCADA systems using the IEC/104 protocol. The research presented in this paper enhances the previous work by deriving a stateful protocol analysis approach to create a stateful Intrusion Detection System (IDS) for IEC/104 SCADA systems.

Stateful protocol analysis is a common and effective detection methodology that operates by comparing predetermined profiles of acceptable protocol behaviors for protocol states against observed activities to detect deviations and misbehaviors [5]. "Stateful" means that the IDS is able to identify and track the states of network, transport, or application protocols that have a concept of state. Stateful protocol analysis of SCADA can be performed in order to ensure proper use of protocols, compliance with protocol standards, and can be used to detect anomalous communications, which may include packet injection, replay attacks and data manipulation. The National Institute of Standards and Technology (NIST) guidelines for industrial control systems recommend stateful inspection for standard IT firewalls to evaluate packet contents at the transport layer. This enables tracking of active sessions in order to determine whether session packets are legitimate. NIST further recommends additional rule sets for SCADA applications [6].

Towards this aim, this paper proposes a model using state machines that analyze and track IEC/104 protocol state transitions in the application layer communications. This

This work was supported in part by the UK EPSRC/RCUK under Grant Number EP/G042594/1, the European FP7 project PRECYSE.

enables validation of communications via the protocol, for the purpose of supporting intrusion detection in the SCADA system.

II. Related Work

Using IDS in SCADA networks is a relatively new concept. Some research has been conducted and applied in intrusion detection approaches for certain SCADA systems, such as signature-based, anomaly-based, model-based intrusion detection methods, as well as other SCADA-specific IDSs [4], [7]-[11]. However, research in this cross-disciplinary cyber-physical context, especially for stateful detection in SCADA systems, still has a long way to go.

Stateful protocol analysis is a typical detection methodology in IT security [5]. Y. Al-Nashif et al. [12] adopt the protocol behavior analysis approach as part of a multi-level IDS, which only describes TCP state behaviors in the transport layer by Finite State Machines (FSM). However, a recent survey has shown that approximately 80% of cyber attacks originate in the application layer [13]. H. Sengar et al. [14] present a protocol state machine based IDS for Voice over IP (VoIP), which identifies any deviation from normal protocol behaviors, and hence, could detect unknown attacks. P. Truong et al. [15] also propose an FSM based intrusion detection model for H.323-based VoIP. In [16], state machines based intrusion detection is proposed for Advanced Metering Infrastructures (AMI) in Smart Grids that includes the device-level state machine for smart meters and the application-level state machine for the American National Standards Institute (ANSI) C12.22 protocol. However, there is little published literature which rigorously considers stateful intrusion detection for SCADA protocols. To this end, by exploiting an in-depth protocol analysis and a DPI method, a stateful intrusion detection approach for IEC/104 driven SCADA systems is proposed in this paper.

III. Stateful Protocol Analysis based Detection

The stateful protocol analysis based detection uses a *whitelist* methodology, which means it identifies any abnormal packets that violate the predefined protocol state behaviors.

In practical Man-in-the-Middle (MITM) attack scenarios [4], it is difficult to completely block normal communication between clients and servers and replace it solely with malicious data. However, it can be relatively simple to inject a false packet into the normal communication connection (e.g., by a TCP hijacking attack). In this context, a stateful IDS must be able to identify any abnormal packet that violates the predefined state diagrams.

Finite State Machines (FSM) provide an effective methodology to describe dynamic behaviors of reactive systems [17]. A communication protocol is one of the common examples of such systems. For example, the IEC/104 protocol can be designed and implemented by an FSM. This paper proposes a novel stateful detection approach using a Detection State Machine (DSM) based on the concept of the FSM.

A. Detection State Machine

Definition 1: A Detection State Machine (DSM) D is a sixtuple:

$$D = (S, \Sigma, G, A, T, S_A) \quad (1)$$

where :

- S is a finite set of protocol states, including the start state and final state.
- Σ is an event alphabet of the DSM.
- G is a set of guard conditions on transitions.
- A is a finite tuple (a_1, a_2, …, a_n) of transition actions, including entry action and exit action.
- T is a transition relation: $S \times \Sigma \times G \rightarrow (S, A)$.
- S_A is a set of alarm states ($s_{a_1}, s_{a_2}, ..., s_{a_n}$) of the DSM.

Each transition t in the transition relation set T ($t \in T$) is a quintuple $<s_s, e, g, s_t, a>$, where s_s, $s_t \in S$ are the source and the target states of the transition respectively, $e \in \Sigma$ is an event that is an input message to the source state, $g \in G$ is a guard condition on the transition which is a Boolean expression that must be true for the transition to be taken, and $a \in A$ defines an action to be performed by the state machine, which is associated with the transition or with entering or exiting a specific state. The basic state transition unit is shown in Fig. 1, which is the fundamental element to constitute specific state machines in practical applications.

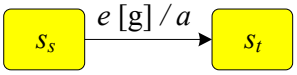

Figure 1. The basic state transition unit.

In comparison with conventional FSMs, the presented DSM introduces the concept of a set of alarm states S_A. When any protocol misbehavior occurs that deviates from the state machines based on the protocol specification, the DSM will be trigged and the state will transfer to an alarm state s_a ($s_a \in S_A$).

B. Proposed Stateful IDS for IEC/104

Following an in-depth analysis of the behaviors of the IEC/104 protocol, a stateful IDS is proposed for intrusion detection in IEC/104 SCADA systems. The main idea is to develop and deploy the stateful IDS to monitor the communication traffic between the IEC/104 client and the server. The stateful IDS uses DFM not only to describe the important and normal protocol behaviors in the form of state transitions, but also to detect protocol misbehaviors in the form of alarm states, as shown in Fig. 2. The detection function is performed by state transition analysis of the DFM. In the detection state diagram, the current state together with the captured and parsed IEC/104 packet will determine the normal state transition (update or keep the current state) or the abnormal alarm state. As a *whitelist* methodology, the DFM based IDS can detect deviations from normal state transitions, and hence, detect unknown attacks.

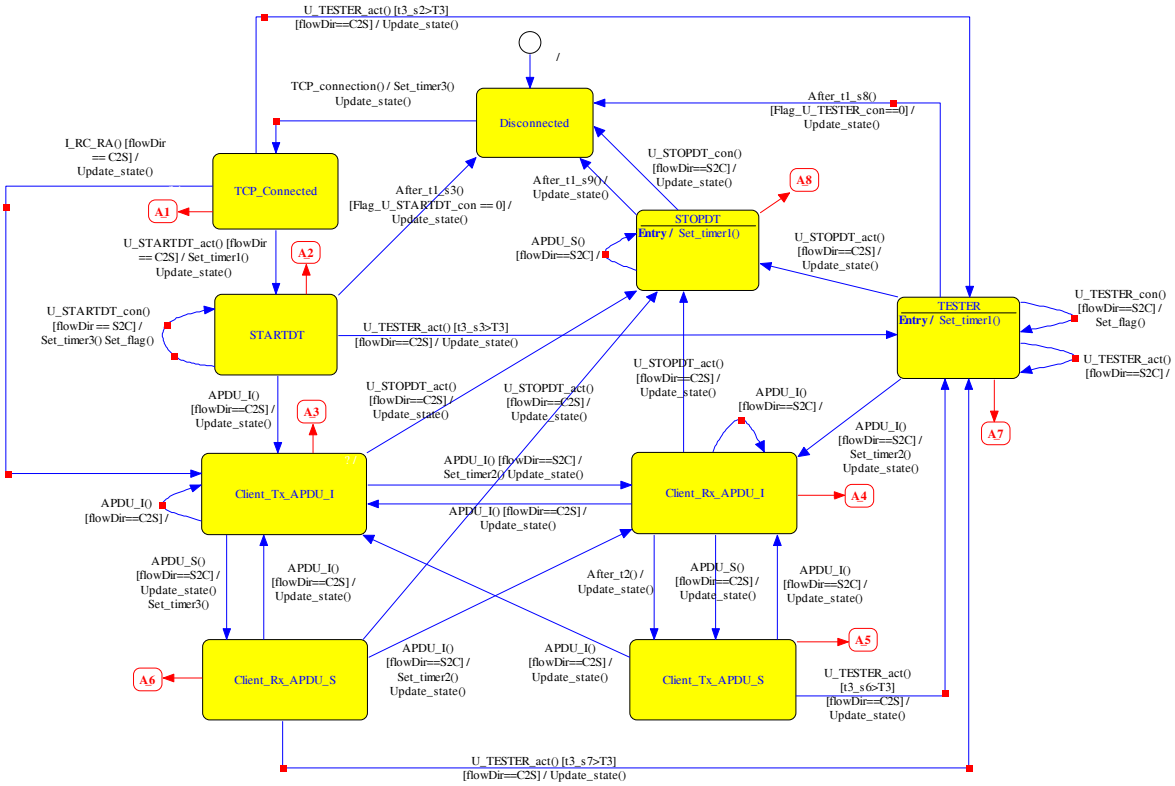

Figure 2. The detection state diagram for IEC/104 stateful IDS.

In Fig. 2, the yellow rectangles represent normal states which are defined according to TCP connection states, APCI information, and the states of the client, as described in Table I. The states named A_1 – A_8 represent the alarm states. The lines with arrows stand for transitions. There are two kinds of events in Fig. 2: packet-capturing events and time events, as shown in Table II. The packet-capturing events are described based on the arrival of packets. For example, an *APDU_I* event means arrival of an *I* format APDU packet. A time event will occur when a timer is overtime. For example, an *After_t2* event is triggered after timer 2 is overtime.

In terms of actions in Fig. 2, *Set_flag* means setting a flag for a guard condition; *Set_timer1*, *Set_timer2* and *Set_timer3* stand for setting timer 1, timer 2 and time 3 for corresponding time event, respectively; *Update_state* is to update a state. The guard conditions are explained in Table III. The constants referred to in the proposed DSM for IEC/104 are described in Table IV.

TABLE I. THE NORMAL STATE TABLE

No.	State name	Remarks
1	Disconnected	No TCP connection between a client and a server
2	TCP_Connected	A TCP connection has been established between a client and a server
3	STARTDT	Start data transfer
4	Client_Tx_APDU_I	A client sends an *I* format APDU
5	Client_Rx_APDU_I	A client receives an *I* format APDU
6	Client_Tx_APDU_S	A client sends an *S* format APDU
7	Client_Rx_APDU_S	A client receives an *S* format APDU
8	TESTER	Periodical connection test procedure
9	STOPDT	Stop data transfer

TABLE II. THE EVENT TABLE

Event name	Remarks
APDU_I	The arrival of an *I* format APDU packet
APDU_S	The arrival of an *S* format APDU packet
After_t1_s3	Time event, after t1_s3 is overtime
After_t1_s8	Time event, after t1_s8 is overtime
After_t1_s9	Time event, after t1_s9 is overtime
After_t2	Time event, after t2 is overtime
After_t3_s2	Time event, after t3_s2 is overtime
After_t3_s3	Time event, after t3_s3 is overtime
After_t3_s6	Time event, after t3_s6 is overtime
I_RC_RA	The arrival of a remote command or remote adjustment packet
TCP_connection	TCP three-way handshake
U_STARTDT_act	The arrival of a START act packet
U_STARTDT_con	The arrival of a START con packet
U_STOPDT_act	The arrival of a STOPDT act packet
U_STOPDT_con	The arrival of a STOPDT con packet
U_TESTER_act	The arrival of a TESTER act packet
U_TESTER_con	The arrival of a TESTER con packet

TABLE III. THE GUARD CONDITION TABLE

Guard condition	Remarks
flowDir == C2S	The flow direction is from a client to a server
flowDir == S2C	The flow direction is from a server to a client
Flag_U_STARTDT_con==0	No receiving a STARTDT con packet
Flag_U_TESTER_con==0	No receiving a TESTER con packet
t3_s2 > T3	Timer t3 of the state 2 is overtime
t3_s3 > T3	Timer t3 of the state 3 is overtime
t3_s6 > T3	Timer t3 of the state 6 is overtime
t3_s7 > T3	Timer t3 of the state 7 is overtime

TABLE IV. THE CONSTANT TABLE

Constant name	Default value	Remarks
C2S	1	From a client to a server
S2C	0	From a server to a client
T0	30 (s)	Time-out of connection establishment
T1	15 (s)	Time-out of send or test APDUs
T2	10 (s)	Time-out for acknowledgements in case of no data messages T2 < T1
T3	20 (s)	Time-out for sending test frames in case of a long idle state

The variables used in Fig. 2 are shown as follows: *Flag_U_STARTDT_con* or *Flag_U_TESTER_con* is a flag of capturing a STARTDT con or TESTER con packet, respectively; *flowDir* and *flowID* mean the direction and the identifier of a flow, respectively; *tm_sn* is timer tm of the state *n*, for example, *t1_s3* means timer t1 of the state 3; *t2* stands for timer t2.

IV. IMPLEMENTATION

Due to the requirements of the DSM, it is not easily feasible to implement using *Snort*. The proposed stateful IDS is therefore implemented using the Internet Traffic and Content Analysis (*ITACA*) tool. *ITACA* [18] is a software platform for traffic sniffing and real-time IP network analysis which has been developed by the Center for Secure Information Technologies (CSIT) at Queen's University Belfast. The extendable and flexible analysis tool enables the implementation of plug-ins to perform specific tasks, e.g., intrusion detection. The detailed modules and functions of the *ITACA* architecture are described in [18]. In this paper, the Stateful Protocol Analysis (SPA) module in the stateful IDS plug-in is developed in C/C++ using the *ITACA* platform, as shown in Fig. 3.

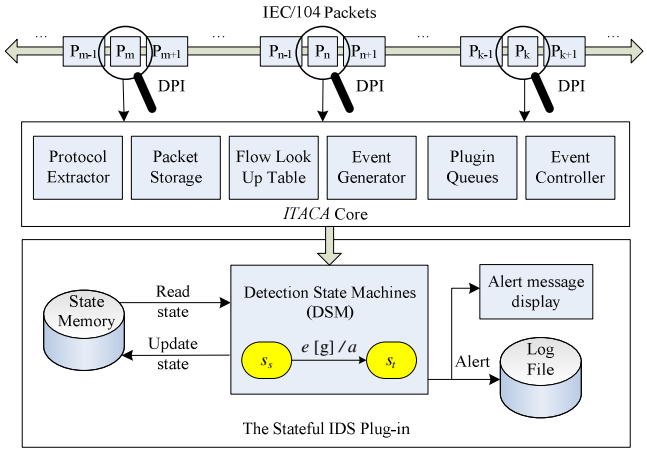

Figure 3. The *ITACA* based implement of the proposed stateful IDS

The detailed implementation steps are as follows.

1) The ITACA core captures, extract, parse and analyse the raw IEC/104 packets in order to provide all possible information for the stateful IDS plug-in.

2) The DSM, as presented in Section III and described in Fig. 2, is implemented within a new stateful IDS plug-in.

3) Based on each captured IEC/104 packet and the current state read from the state memory, the DSM will determine whether the state transition is normal or abnormal. If it is abnormal, the stateful detection module will generate an alert notification and record the detection result in the log file.

For example, when the current state is `TCP_Connected`, the DSM will execute, as illustrated by part of the DSM program shown in Fig. 4.

```
CurrentState = StateMemory.state;
switch (CurrentState)
{
  case TCP_Connected:
  {
    if((packet->payload == U_STARTDT_act)&&(packet
     ->flowDir == C2S))
    {
      StateMemory.state = STARTDT;
      t1_s3 = packet->packet_time;
    }
    else if((packet->payload == I_RC_RA)&&(packet
    ->flowDir == C2S))
    {
      StateMemory.state = Client_Tx_APDU_I;
    }
    else if ((packet->payload == U_TESTER_act) &&
    (packet->flowDir == C2S) && (t3_s2 > T3))
    {
      StateMemory.state = TESTER;
      t1_s8= packet->packet_time;
    }
    else
    {
      alert();
    }
  }
  break;
  …
}
```

Figure 4. A portion of the pseudo codes of the stateful detection module

First, the current state (`TCP_Connected`) is read from the state memory (`StateMemory.state`).

Second, the DSM executes switch statements to determine the state transition according to the arrived packet and the current state. If the captured packet is a START act packet (`packet->payload==U_STARTDT_act`) and the flow direction is from the client to the server (`packet->flowDir==C2S`), the state memory is updated to the state 3 (`STARTDT`) and meanwhile the timer t1 of the state 3 is set. When the arrived packet is a remote command or remote adjustment packet (`packet->payload==I_RC_RA`) in the control direction (`packet->flowDir==C2S`), the updated state is `Client_Tx_APDU_I`. If a TESTER act packet (`U_TESTER_act`) in the control direction arrives and the timer t3 of the state 2 is overtime (`t3_s2 > T3`), the next state is the state 8 (`TESTER`) and the timer t1 of the state 8 is set.

Finally, other misbehavior packets will call the alert function that generates and records the alarm messages.

In all, 8 stateful detection rules for IEC/104 are implemented in the proposed stateful IDS.

V. EXPERIMENTAL RESULTS

A. *Experiment*

An experimental process was developed as follows.

1) The normal IEC/104 traffic was obtained from the commercial source (a mirrored SCADA test-bed in a power company). Abnormal packets were generated by modifying the captured data or by injecting new malicious packets in the pre-captured Packet Capture (PCAP) file.

2) The PCAP file was read by *ITACA* core that extracts and interprets all available information for the SCADA-IDS plug-in. The IEC/104 packets were monitored and detected by the developed stateful IDS plug-in, as mentioned in Section III.B.

3) The detection results were displayed and recorded into a log file.

B. Results

In the experiment, there were 116497 packets with 28 abnormal packets in the PCAP file, and wherein the number of abnormal packets violating SPA was 28. It is apparent from the experimental results that the proposed stateful IDS effectively identifies all the abnormal data with zero false positive for the given deterministic rules.

The message format in the log file is defined referring to RFC 3164 as follows:

`<SEVERITY> TIMESTAMP DEVICE_NAME DEVICE_TYPE ALERT_TYPE EVENT_DESCRIPTION SRC_IP SRC_PORT DST_IP DST_PORT`

In this case `SEVERITY` represents alert severity which is described by numerical code, e.g., `0`, `1` and `2` stand for HIGH, MEDIUM and LOW, respectively. The `TIMESTAMP` field is the local time and is in the format of "`YYYY-MM-DD HH:MM:SS`". `DEVICE_NAME` means the name or IP address of specific security device. `DEVICE_TYPE` is the type of the security device, e.g., `IDS`. `ALERT_TYPE` represents alert event type which is violated such as `SPA`. `EVENT_DESCRIPTION` describes the detailed information of specific security event. `SRC_IP`, `SRC_PORT`, `DST_IP` and `DST_PORT` are source IP address, source port, destination IP address and destination port, respectively.

As described in Fig. 5, the SCADA-IDS alert messages show that abnormal packets in the states `TCP_Connected`, `STARTDT` and `Client_Tx_APDU_I` are identified, which violate the detection rules SPA-1, SPA-2 and SPA-3, respectively. Other detection results have also been recorded in the log file using the unified format.

```
<0> 2013-10-08 21:11:34 SCADA-104-IDS IDS SPA-1
Suspicious packet from the state TCP_Connected
10.1.8.13 b997 10.6.5.158 964

<0> 2013-10-08 21:15:12 SCADA-104-IDS IDS SPA-2
Suspicious packet from the state STARTDT
10.1.8.13 b997 10.6.5.158 964

<0> 2013-10-08 21:18:42 SCADA-104-IDS IDS SPA-3
Suspicious packet from the state Client_Tx_APDU_I
10.1.8.13 b997 10.6.5.158 964
```

Figure 5. Alert examples in the log file

VI. CONCLUSION

Previous research in this area has mainly investigated Modbus or DNP3 protocols [7]-[9]. To the best of the authors' knowledge, this paper is the first to propose a stateful IDS for IEC 60870-5-104 SCADA networks. A novel detection state machine is proposed and applied to provide the stateful IDS for SCADA networks using IEC/104. The proposed intrusion detection tool, implemented by the *ITACA* platform, can be applied to monitor and detect IEC/104 traffic in SCADA systems. Stateful protocol analysis is recommended by NIST to detect anomalous behaviors where SCADA protocols are used. This research has directly addressed these challenges through the presented stateful IDS. This work contributes significantly to improving the cyber-security of SCADA systems that use the IEC/104 protocol.

REFERENCES

[1] R. Langner, "Stuxnet: Dissecting a Cyberwarfare Weapon," *IEEE Security & Privacy*, vol. 9, pp. 49-51, May 2011.

[2] *Telecontrol Equipment and Systems—Part 5-104: Transmission Protocols—Network Access for IEC 60870-5-101 Using Standard Transport Profiles,* IEC Standard 60870, 2006.

[3] *IEC Telecontrol Equipment and Systems—Part 5-101: Transmission Protocols—Companion Standard for Basic Telecontrol Tasks,* IEC Standard 60870, 2003.

[4] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, B. Pranggono, and H. F. Wang, "Intrusion Detection System for IEC 60870-5-104 based SCADA networks," in *Proc. 2013 IEEE Power and Energy Society General Meeting*, pp. 1-5.

[5] K. Scarfone and P. Mell, "Guide to intrusion detection and prevention systems (IDPS)," Special Publication 800-94, National Institute of Standards and Technology (NIST), Gaithersburg, MD., Feb. 2007.

[6] K. Stouffer, J. Falco, and K. Kent, "Guide to Industrial Control Systems (ICS) Security – Recommendations of the National Institute of Standards and Technology," Special Publication 800-82, NIST, Gaithersburg, MD., Jun. 2011.

[7] Quickdraw SCADA IDS. [Online]. Available: http://www.digitalbond.com

[8] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for SCADA networks," in *Proc. 2007 the SCADA Security Scientific Symposium*, pp. 127–134.

[9] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. N. Fovino, and A. Trombetta, "A Multidimensional Critical State Analysis for Detecting Intrusions in SCADA Systems," *IEEE Trans. Industrial Informatics*, vol. 7, pp. 179-186, May. 2011.

[10] B. Zhu and S. Sastry, "SCADA-specific Intrusion Detection/Prevention Systems: A Survey and Taxonomy," in *Proc. the First Workshop on Secure Control Systems*, pp. 1-16, 2010.

[11] S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber-Physical System Security for the Electric Power Grid," *Proc. IEEE*, vol. 100, no. 1, pp. 210-224, Jan. 2012.

[12] Y. Al-Nashif, A.A. Kumar, S. Hariri, Q. Guangzhi, L. Yi, and F. Szidarovsky, "Multi-Level Intrusion Detection System (ML-IDS)," in *Proc. Int'l Conf. on Autonomic Computing*, pp. 131-140, 2008.

[13] A. Anitha and V. Vaidehi, "Context based Application Level Intrusion Detection System," in *Proc. Int'l conf. on Networking and Services*, pp. 16-21, 2006.

[14] H. Sengar, D. Wijesekera, H. Wang, and S. Jajodia, "VoIP Intrusion Detection Through Interacting Protocol State Machines," in *Proc. Int'l Conf. on Dependable Systems and Networks*, pp. 393-402, 2006.

[15] P. Truong, D. Nieh and M. Moh, "Specification-based intrusion detection for H.323-based voice over IP," in *Proc. 5th IEEE Int'l Symp. on Signal Processing and Information Technology*, pp. 387-392, 2005.

[16] R. Berthier and W.H. Sanders, "Specification-Based Intrusion Detection for Advanced Metering Infrastructures," in *Proc. IEEE 17th Pacific Rim Int'l Symp. on Dependable Computing*, pp. 184-193, 2011.

[17] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of computer programming*, vol. 8, pp. 231-274, 1987.

[18] J. Hurley, A. Munoz, and S. Sezer, "ITACA: Flexible, Scalable Network Analysis," in *Proc. IEEE Int'l Conf. on Communications Industry Forum & Exhibit.*, pp.1084-1088, 2012